
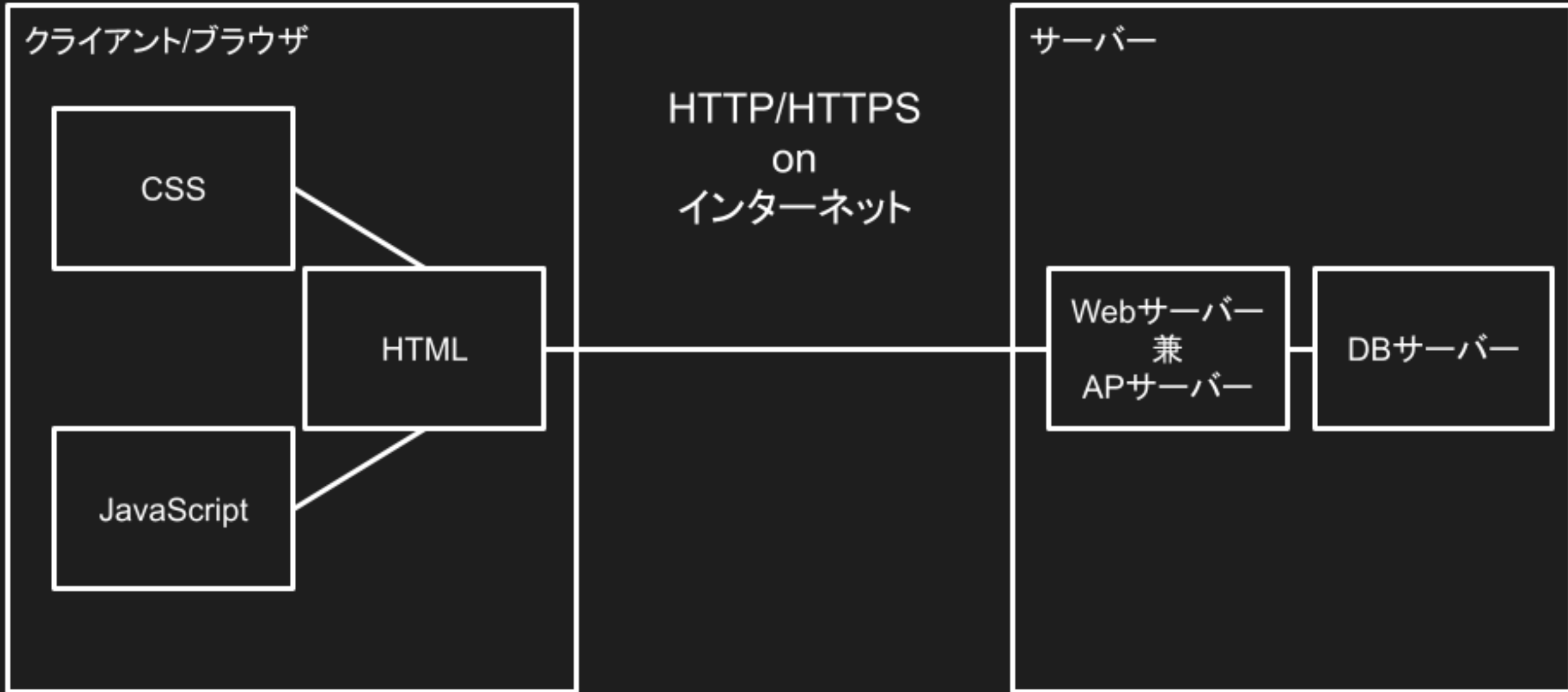


Webアプリケーション入門 🚀

CSSの基本

はじめに：今日やること

- 今日のゴール: 
 - CSSとは何かを知る
 - CSSの基本的な書き方を知る
 - CSSを使ってWebページの見た目を整える



まずはHTMLから

- Webページの「**骨組み**」や「**内容**」を決める言語
- 文章、見出し、画像などを配置する

例えるなら…

家の**骨組み**や**間取り** 🏠









次にCSSの登場！

- HTMLで作った部品（要素）の**「見た目」を整える**言語
- 色、大きさ、配置、余白などを指定する

例えるなら…

家の内装や外装（壁紙、家具、ペンキの色） 🎨

CSSでできること（ほんの一部）

-  色を変える (文字、背景)
-  大きさを変える (文字、画像)
-  配置を決める (右寄せ、中央揃え、横並び)
-  背景に画像を入れる
-  枠線をつける
-  余白を調整する
-  スマホ対応 (レスポンシブデザイン)
-  アニメーションをつける

なんでCSSを使うの？ 🤔

大きく分けて **3つの理由** があります！

1. わかりやすさ ✨
2. 効率性 🚀
3. 表現力 🎨

理由1：わかりやすさ ✨

- HTML（骨組み）と CSS（見た目）を役割分担させる
- コードがスッキリして管理しやすくなる！

例：

- 家の設計図 (HTML) と インテリア指示書 (CSS) を分ける感じ。
- ごちゃ混ぜだと、後でどこを直せばいいか大変💧

理由 2：効率性 🚀

- サイト全体のデザインを一箇所で管理できる！
- 修正がめっちゃくちゃ楽になる！

例：

- 「サイト内の見出しは全部 青色・太字」というルールをCSSファイルに書く。
- → そのCSSファイルを変更するだけで、**全ての見出しが一瞬で変わる！**
- （HTMLに直接書いてたら、全ページ修正…🤖）

理由3：表現力 🎨

- HTMLだけでは難しい、凝ったデザインが実現できる！
- 自由なレイアウトが可能に！

例：

- 要素を横に並べる
- 要素を重ねる
- 動きをつける (アニメーション)
- スマホやタブレットなど、画面サイズに合わせた表示 (レスポンシブデザイン)

まとめ

- **HTML**：Webページの骨組み・内容
- **CSS**：Webページの見たい目・デザイン

どうやって書くの？ 簡単なルール

CSSは「誰に」「何を」「どうするか」を指定します。

```
セクタ {  
  プロパティ: 値;  
}
```

- **セクタ: 誰に** (どの部分に) スタイルを適用するか
 - 例: `h1`, `p`, `.classname`
- **プロパティ: 何を** (どの見た目を) 変えたいか
 - 例: `color`, `font-size`
- **値: どういう風**に 変えるか
 - 例: `blue`, `16px`

どうやって書くの？ (具体例)

```
/* h1(見出し1) の 文字色(color) を 青色(blue) にする */  
h1 {  
  color: blue;  
}  
  
/* p(段落) の 文字の大きさ(font-size) を 16ピクセル(16px) にする */  
p {  
  font-size: 16px;  
}  
  
/* h1 に複数のスタイルを適用 */  
h1 {  
  color: blue;          /* 文字色を青に */  
  text-align: center;   /* 文字を中央揃えに */  
}
```

{ } の中に、変えたいことをいくつも書けます！ (各指定の終わりは **;**)

どこに書くの？ (3つの場所)

CSSを書く場所は主に3つあります。

1. おすすめ！ 外部ファイル (外部スタイルシート) ✨
2. HTMLの中 (内部スタイルシート)
3. 要素に直接 (インラインスタイル)

まずは「外部ファイル」に書く方法を覚えましょう！

どこに書くの？ ① 外部ファイル (一番おすすめ！)

- `style.css` のような **CSS専用ファイル** を作る。
- CSSのルールをそのファイルに書く。
- HTMLファイルの `<head>` タグ内で、`<link>` タグを使って読み込む。

```
<head>  
  <link rel="stylesheet" href="style.css">  
</head>
```

- **メリット:**
 - HTMLがスッキリ ✨
 - 複数のページで使い回せる ♻️
 - 修正が楽 (CSSファイルを直せば全ページに反映) 🛠️

どこに書くの？ ② HTMLの中

② HTMLの中 (内部スタイルシート)

- HTMLファイルの `<head>` 内に `<style>` タグで囲んで書く。

```
<head>
  <style>
    p { color: green; }
  </style>
</head>
```

- そのHTMLページ**だけ**に適用したいスタイルに。

どこに書くの？ ③ 要素に直接

③ 要素に直接 (インラインスタイル)

- HTMLタグに直接 `style="..."` と書き込む。

```
<p style="color: red; font-size: 12px;">小さい赤い文字</p>
```

- ちょっとした調整には便利だけど、多用すると管理が大変に… 💧

「誰に」伝える？ (セレクタの基本)

どの部分の見た目を変えるか指定する方法です。

- **タグ名:** `h1`, `p`, `div` など
 - その種類のタグ **すべて** に適用
 - `p { color: gray; }` /* すべての段落を灰色に */

「誰に」伝える？ (セレクタの基本)

- クラス (あだ名): `.classname` (ドット `.` + 名前)
 - HTML側: `<p class="important">大事！</p>`
 - CSS側: `.important { color: red; }`
 - **複数の要素に同じスタイル** を適用したい時に超便利！
- ID (名前): `#idname` (シャープ `#` + 名前)
 - HTML側: `<div id="header">ヘッダー</div>` (IDはページ内で**1つだけ**)
 - CSS側: `#header { background-color: lightblue; }`
 - **特定の場所** をピンポイントで指定する時に。

「何を」「どうする」？ (よく使うプロパティ)

CSSで見た目を変更する命令の一部です。

- `color` : 文字の色
- `background-color` : 背景の色
- `font-size` : 文字の大きさ (例: `16px` , `1.5em`)
- `font-family` : フォントの種類 (例: `sans-serif` , `"メイリオ"`)
- `font-weight` : 文字の太さ (例: `bold`)

よく使うプロパティ (続き)

- `text-align` : 文字の揃え方 (例: `center`)
- `width` / `height` : 幅 / 高さ
- `margin` : 外側の余白
- `padding` : 内側の余白
- `border` : 枠線 (例: `1px solid black`)

どうやって勉強する？

1. まずはやってみる！

- 簡単なHTMLとCSSファイルを作る。
- 値を色々変えて、ブラウザで表示確認！（F5 で更新）

2. 真似して、変えてみる！

- Web上のサンプルコードを試す。
- 色やサイズを変えて実験！

3. 困ったら調べる！

- MDN Web Docs (超おすすめ！) ✨ developer.mozilla.org/ja/docs/Web/CSS
- 検索エンジン: 「CSS 文字色 変え方」 など

4. ブラウザの開発ツール (F12)

- Webサイトの裏側を覗いたり、自分のCSSを確認したりできる便利ツール！

まとめ

- CSSはWebページの「見た目」を作る言語！
- 基本ルール: セクタ { プロパティ: 値; }
- 外部ファイル (`.css`) に書くのがおすすめ！
- `color` , `font-size` , `margin` , `padding` などよく使うものから覚えよう！
- 習うより慣れろ！ まずは簡単なことから試してみましょう！